

Complex Adaptive Systems, Volume 1
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2011- Chicago, IL

A multiagent framework for industrial robotic applications

Marko Švaco^{a*}, Bojan Šekoranja^a, Bojan Jerbić^a

^a*Faculty of Mechanical Engineering and Naval Architecture, Ivana Lučića 5, 10000 Zagreb, Croatia*

Abstract

The paper presents a novel approach toward modeling and governing complex system behavior in flexible and adaptive robotic assembly systems. A fully distributed multiagent approach is implemented for autonomous control. The system is defined at multiple levels of granularity where agents provide services in respect to the current global goal. A decentralized multiagent approach is adopted for reasons of flexibility and fault tolerance embedded in the design phase. To prove the concept a robotic application for intelligent assembly is presented and discussed. It consists of multiple industrial robots equipped with force/torque sensors, 2D and 3D vision systems, automatic tool changers and other sensors and actuators. Through fusion of sensory input and mutual communication agents construct and negotiate an assembly plan and reconfigure respectively.

© 2011 Published by Elsevier B.V.

Keywords: Distributed robotic systems; Multiagent systems; Assembly technology

1. Introduction

Dynamic global economy significantly affect the technological changes in the industrial assembly. Production is oriented toward small quantities and numerous variants of products tailored for specific customer needs. Mass customization [1] is getting more important than mass production. Highly responsive [2] and flexible systems need to replace single-purpose machines and production lines that address only specific products. The system hierarchy and control methods need to adapt to these requirements. Therefore, traditional automation becomes inefficient and expensive for today's industrial expectations. Manufacturers are facing the ever growing problems of reduction in batch sizes and constant requirements in product variations. Multiagent systems [3] exhibit characteristics that are beneficial and applicable in such conditions. Inherently distributed [4], with the property of operating without the

* Corresponding author. Tel.: +385 1 6168 423; fax: +385 1 6156 940.

E-mail address: marko.svaco@fsb.hr

need for central control; and self-organization, are main attributes that can be utilized for control of flexible and adaptive production and assembly systems. Due to their open and dynamic nature robot architectures based on agent concept seem to be suitable frames to respond to uncertainties without explicitly programming every solution to them. The assembly technology has been stressed out as the main contributor to overall system efficiency in a product lifecycle.

Related works have addressed a wide range of applications in the domain of multiagent industrial systems. In [5] new and complex actions are the result of emergent behavior and self-organization in multi-agent assembly systems. In the research carried out in [6] a methodology for fault tolerant design of a multiagent system has been presented and verified. Resource scheduling [7] is also one multiagent industrial application that is very common and utilizes the appropriate properties of agents. The use of the FIPA [8] agents has allowed researchers to create adaptive multiagent systems for a wide range of applications. All these works have shown that the agent architecture is a promising approach and is capable of addressing issues in non-deterministic systems and applications. The main drawback of the mentioned works is its' operating cycle time which is not suitable for real time industrial control.

This work focuses on an actual industrial setup and process and the discussed topic has been implemented as a main control frame for robotic assembly of real electronic products. All equipment used in the presented research is standardized industrial hardware and software. This is primarily important for ensuring robust hardware architecture in terms of operating conditions and reliability. Limited processing power, memory and storage space are the real conditions that have properly been addressed. The system incorporates all major industrial control principles translated to the level of universal applicable control mechanisms for the agent architecture. Prior research steps have been verified in automatic planning [9] and development of the robotic assembly workcell [10].

The paper is structured as follows. In Section 2 the main notions and principles of the multiagent architecture are presented. In Section 3 the calibration method for multi-robot interaction is detailed. Agent communication and negotiation schemes are discussed in Section 4. Section 5 presents the application of the architecture on an industrial multi robot setup. Finally, the paper is concluded in Section 6.

2. The multiagent architecture

Traditional assembly systems are unreliable, inflexible and with no or limited space for adaptive control [11] and reconfiguration. Hardcoded PLC control routines provide a rigid and reliable solution in aspects of mass production and predictable assembly conditions. The bottleneck toward flexible and adaptive control is the centralised control architecture [12]. For uncertain and unpredictable assembly conditions the process of mapping all possible states of the environment in a central device presents a weak point. The memory and processing power of the controlling unit is limited therefore posing a reduced scope of operation and control power. Distributing the computation and other tasks among individual agents or groups of agents is the first step. The developed multiagent architecture has been tested and verified on an actual industrial setup consisting of 8 industrial robots, 4 transport systems, 7 vision systems, 2 F/T sensors, 3 PLC's and automatic tool changers. Supported with essential communication protocols such as DeviceNet, Profibus, and TCP/IP allows integration of diverse technical components into the system. By introducing a certain level of autonomy to each individual component of the system overall capabilities increase. Each component (agent) has the ability of perceiving its' environment with various type of sensors and extracting meaningful information from it. The organization of agents is based on the assembly task and can range from a fully distributed architecture to master-slave configurations in multi-arm part manipulation. More processing power is obtained from delegating tasks to individual entities inside the assembly system and making the process planning and decision making local at the agent side. Following these assumptions each mechanical robot unit has its' delegated controller. Multiagent organizational structure of all working elements is the assumption for dynamic and autonomous reconfiguration of the system based on input parameters gathered by the agents. In assembly systems the main part of each process are handling operations, proper positioning and joining of parts in more complex entities - products.

3. Multi-robot calibration

An agents' task is to assemble a product that is built from various objects with defined relationships. A product $\mathbf{P} = \{\mathbf{Q}_i \mathbf{b}_{j,k}\}$ is a set of relations \mathbf{Q}_i ($i = 1 \dots m-1$) between objects $\mathbf{b}_{j,k}$ ($j = 1 \dots n, k = 1 \dots u$). The multiagent system has

properties of a market organization type [13], [14] where agents bid [15] for given resources (parts) in their workspace. Based on the actual configuration, processes queues agents place their bid on the multiagent virtual blackboard (MVB). Time schedules need to be negotiated when areas of interest in the global workspace are not occupied. Global goal G is the actual product that must be assembled from available elements following the given set S . Agents in the system are collaborative and tend to optimize the global system performance in order to provide higher possible efficiency. The efficiency is measured in consumed resources (time, energy and path) and parts assembled in a defined time period. Certain assembly operations need to be done in a multi-robot fashion where the degrees of freedom excreted by one robotic arm are insufficient. Multi-robot operation requires a precise calibration method. Cumulative absolute accuracy of multiple robotic units is an issue that needed to be addressed for several reasons which include:

- Robust operation
- Precise assembly tasks of multiple robots
- Task planning regarding space collisions in shared agent workspace

A two-step calibration method was developed for precise positioning of multiple robots in a particular shared workspace quadrant. First the coarse spatial calibration of multiple robots R_n ($n = 1...a$) was performed. Robot tool center points (TCP) were guided to a desired position in the shared workspace. A set of global Cartesian coordinate systems C_m ($m = 1...b$) were acquired with the following parameters: $C_m = \{K_1, K_2...K_n\}$. K_n depicts a set of local coordinates accessible in C_m to the current robot. Obtained absolute accuracy results with a lower order error than a single robotic unit and therefore cannot provide desired precision for assembly. Calibration information is written as global knowledge for each robot and is used in the visual calibration step. A schematic view of the visual method for error correction and precise positioning is depicted in Fig 1.a) Shared robot workspace is divided in spatial quadrants with inherent errors provided by the initial robot calibration (Fig 1. b). Each quadrant is a cube of side length 100 mm. In these quadrants the coarse positioning of robot TCP's is achieved. Accurate positioning is established using visual feedback by acquiring relative positions of robot TCP's using markers. Points P and P' reached by *Robot1* and *Robot2* respectively are depicted in Fig 1. b) The initial offset $\Delta_{xyzx'y'z'}$ is the result from initial imprecise calibration. *Robot2* visually identifies the relative position of *Robot1* and stores the information. A new coordinate system is identified with an offset about the initial system. The spatial calibration of two robots for the current quadrant is subsequently written in a 3D matrix as a correction index $CI_{p,q}$.

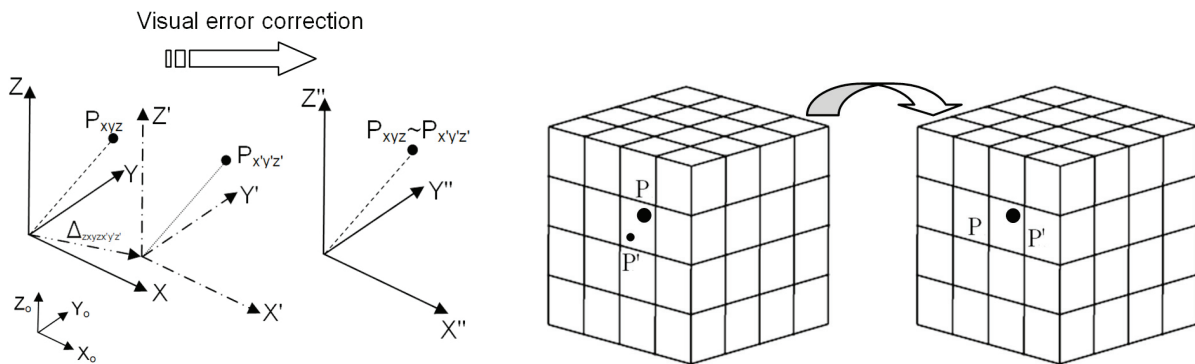


Fig. 1. a) Multi-robot visual calibration method b) Workspace quadrants

4. Multiagent communication protocols

For explicit multiagent robot programming a service oriented multiagent architecture was modeled. System components i.e. agents are self-aware entities capable of decision making and negotiation. Tending towards a common global goal, all agents in the system constantly communicate and negotiate assembly actions. The MVB is used as the main service for the negotiation and task delegation for multiagent communication. Communication is defined at multiple levels of granularity as depicted in Fig 2. The MVB is at the top communication level ensuring that all agent requests are sent to the right participants of the multiagent system. The MVB utilizes fast industrial

protocols for direct binary communication which ensures system stability and uptime. At the second level are peer-to-peer (PTP) communication protocols through which the agents communicate directly. This communication channel is delegated to a pair of agents which have common interests and need to negotiate service requests, free common workspace, visual calibration, etc. The data sent through this type of connection can be of any type including digital signals, process data, service oriented information, vision image data and etc. Agents can form coalitions (groups of agents) if some task cannot be performed individually. The group communication is delegated for this type of architecture and is at the third level. Agents communicate through a delegated group channel (Fig 2.) not connected with other participants of the system. In a master-slave hierarchy group broadcast (GB) messages are sent by the master agent. These messages can include common workspace positions, delegation of tasks to slave agents and etc.

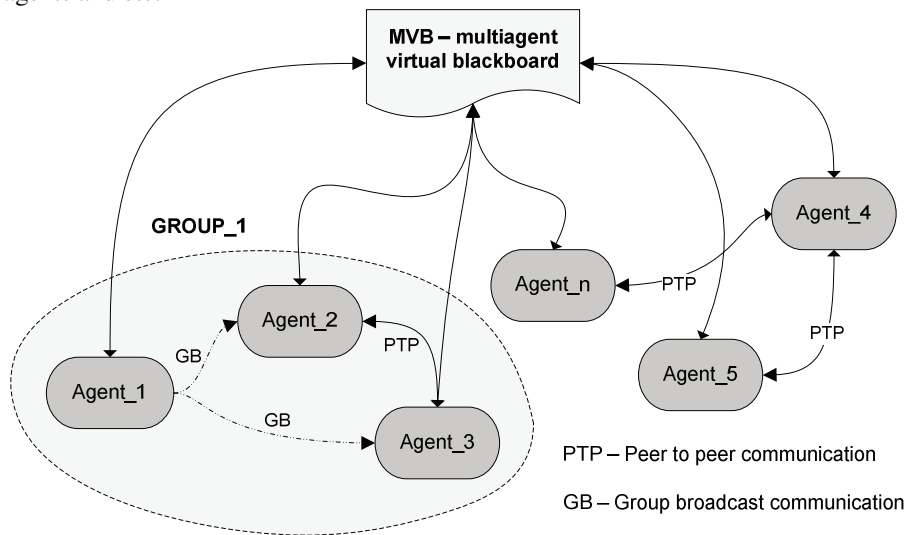


Fig. 2. Communication in the multiagent system

By following the general assembly plan (GAP) given in Fig 3, agents are familiarized with all relevant assembly information. The GAP is situated on the MVB from where the agents download the assembly plan locally. The GAP comprises part, product and assembly process information. The assembly sequence is written as a set of abstract steps that can be translated into specific tasks within a single agent's plan. The GAP is a notation of the assembly sequence and does not take into consideration a particular agent for providing a service or accomplishing a particular task. Therefore the GAP is encoded in a comprehensible way and can be interpreted for any given agent in any given initial state. An agent has local knowledge of its capabilities and compares them with requests from the GAP. By inspecting their actual state, the state of the environment and current process stage agents reason about further necessary actions. Tasks can be performed either individually or by requesting other agents through PTP communication protocols or by coalition forming.

The set of multiagent services include:

- Pick (pick_position, C_m)
- Place (place_position, C_m)
- Hold (hold_position, C_m)
- Transport (initial_position, final_position, C_m)
- Reorient (initial_orientation, final_orientation, C_m)
- Assemble (assembly_position, assembly_operation, assembly_parameters, C_m)
- Inspect (inspect_position, inspect_parameters, camera_parameters, C_m)

Through these services new global behaviors emerge. If a robot needs a specific part defined in the GAP and that part is not currently in its workspace it requests this part using the *Transport* service.

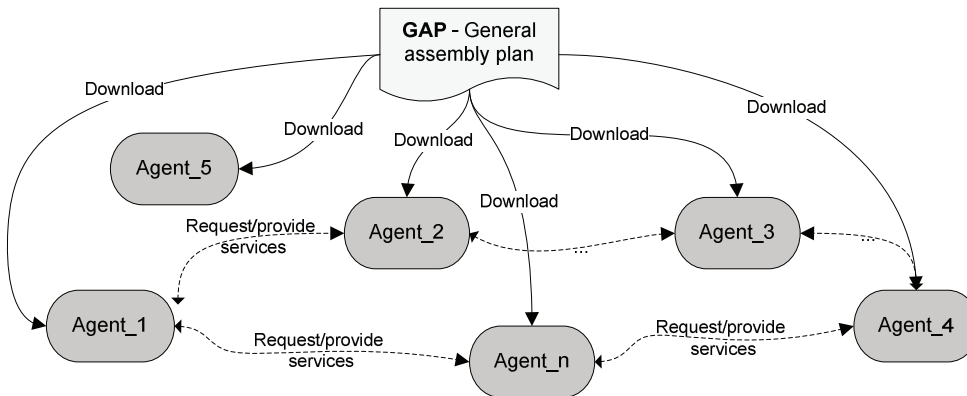


Fig. 3. General assembly plan and agent service architecture

A robot that can transport the part places a bid on the MVB. The bid usually comprises of resources necessary for providing a particular action. The agent with the lowest bid: $b(\text{time}, \text{path_parameters}, \text{additional_parameters})$, is delegated the assembly operation. The actions defined in the service request are performed and other agents are notified of the accomplished assembly task from the GAP.

5. Implementation

The framework has been tested on an actual system consisting of four 6 DOF (degree of freedom) industrial robots, two 4 DOF robots, a 2 DOF robot and a 7 DOF robot. In Fig 4. schematics of the assembly system is presented (including robot work envelopes).

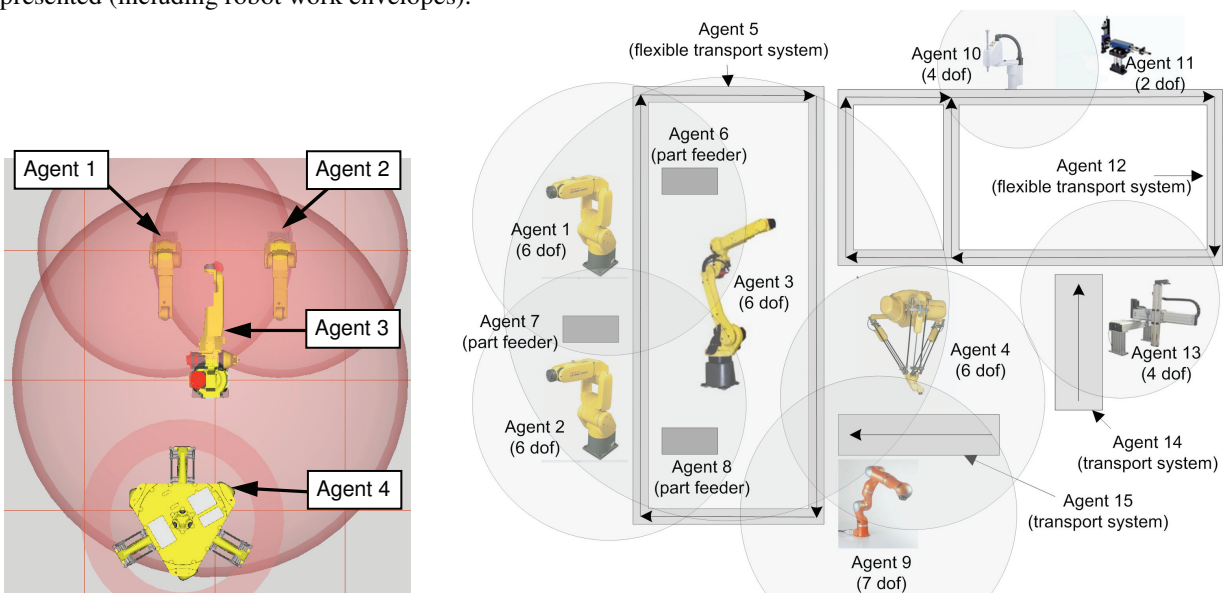


Fig. 4. (a) Part of the actual multiagent robotic system – virtual representation; (b) Global system structure – schematic preview

All robots are equipped with vision systems and automatic tool changers. Additionally two robots (*Agent1* and *Agent2*) have Force/Torque (F/T) sensors for sensitive and high precision assembly operations. All units are oriented around four transport systems that provide (input) base parts and deliver (output) assembled products. A transport system is modeled as an agent which communicates with robotic units through the MVB with binary input and output signals. It can adapt its' speed and type of part carriers in term of current requests from other agents. The preview of the system showing all components was initially modeled in virtual reality along with particular robot

work envelopes. This was done in order to virtually test the layout for a purpose of avoiding unnecessary costs. In this workspace calibration steps presented in Section 3 are performed prior to any multi-robot service realization in a non calibrated quadrant.

For flexible assembly operations automatic tool changers provide additional functionality. Through a synergy of sensorial input in terms of visual data, force feedback, various simple sensors and communication with other agents a decision for appropriate grippers (tools) is determined. Regarding current robot positions, configurations and currently available global information agents organize themselves toward the current assembly activity. A visual identification method is used for acquiring information about respective tool locations.

6. Conclusion

Conventional assembly system organization is mostly oriented toward a single product or a small number of variants. Robot behavior in most of today's industrial environments is controlled from the classical aspect of automatic control. A central system controller governs the assembly process and autonomy within groups of robots or at the level of an individual robot is strictly limited. Small batches of diverse products require an adaptive and flexible system setup as presented in this research. The developed approach ensures a wide-range potential field of applications not bounded by particular parts. Hardware complexity in terms of various 2D and 3D vision systems, F/T and other simple sensors allows adaptive behavioral patterns when encountering uncertainties. As mentioned the support of communication platforms such as DeviceNet, Profibus, and TCP/IP allows integration of diverse technical components into the system forming an open architecture. The system can assemble the current products and with new demands reconfigure and assemble an entirely different product. Explicit multiagent programming based on services enables programming of basic assembly structures (pick, place, hold, transport, etc.). Services are comprehensible for operators where programming the system for new tasks is not time consuming.

References

1. H. L. Hales (1992) Automating and Integrating the Sales Function: How to Profit From Complexity and Customization. Enterprise Integration Strategies, Vol. 9, No. 11, 1–9.
2. I. Seilonen, R. Pirttioja and K. Koskinen (2009) Extending process automation systems with multiagent techniques. Engineering Applications of Artificial Intelligence, Vol. 22, Issue 7, (October, 2009), 1056–67, ISSN: 0952-1976
3. M. Wooldridge (2002) *An Introduction to Multiagent Systems*, John Wiley & Sons, 047149691X, Chichester, England
4. Y. Shoham and K. Leyton-Brown (2009) *Multiagent Systems: algorithmic, game-theoretic and logical foundations*, Cambridge Uni.
5. R. Frei, S. G. Di Marzo Serugendo and J. Barata (2008) Designing self-organization for evolvable assembly systems. *Proceedings of Int. Conf. on Self-Adaptive and Self-Organizing Syst. (SASO)*, Venice, 97–106, IEEE
6. L. Ribeiro, J. Barata, B. Alves and J. Ferreira (2011) Diagnosis in Networks of Mechatronic Agents: Validation of a Fault Propagation Model and Performance Assessment, *Proc. of The 2nd Doc. Conf. on Computing, Electrical and Industrial Syst., Caparica, Portugal*, Luis M. Camarinha-Matos (Ed.), 205–214, Springer
7. T. Pirttioja, A. Pakonen, I. Seilonen, A. Halme and K. Koskinen (2005) Multi-agent based information access services for condition monitoring in process automation. *Proc. of The 3rd Int. IEEE Conf. on Industrial Informatics (INDIN 2005)*, Perth, Australia, IEEE
8. <http://www.fipa.org/>; accessed May 18, 2011.
9. T. Stipanovic and B. Jerbic (2010) Self-adaptive Vision System, *Emerging Trends in Technological Innovation*, In: Camarinha-Matos L. M., Pereira P., Ribeiro L., 195–202, Springer, Boston
10. M. Svaco, B. Sekoranja and B. Jerbic (2011) Autonomous Planning Framework for Distributed Multiagent Robotic Systems, *Proc. of The 2nd Doc. Conf. on Computing, Electrical and Industrial Syst., Caparica, Portugal*, Luis M. Camarinha-Matos (Ed.), 147–154, Springer
11. M. Svaco, B. Sekoranja and B. Jerbic (2010) A Multiagent Approach for Development of a Flexible and Adaptive Robotic Assembly Work Cell, *Proc. of The 3rd Int. Conf. on Computational Intelligence and Industrial Application*, Zhang, Y. ; Tan, H. (Ed.), 64–7, IEEE
12. H. P. Tang and T. N. Wong (2005) Reactive multi-agent system for assembly cell control, *Robotics and Com.-Integrated manuf.* 21, 87–98
13. T. Sandholm (1993) An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations, *Proc. of the 11th Conference on AI*, 256–262,
14. M. Schumacher (2001) *Objective Coordination in Multi-Agent System Engineering*, Springer, New York
15. F.S. Hsieh (2006) Analysis of contract net in multi-agent sys., *Automatica* 42, 733–40